REMARKS

Applicants appreciate the Examiner's thorough consideration provided the present application. Claims 1-42 are now present in the application. Claims 1, 3-6, 10, 15-17, 20, 22-25, 27 and 30-39 have been amended. Claim 1 is independent. Reconsideration of this application, as amended, is respectfully requested.

Information Disclosure Citation

Applicants thank the Examiner for considering the references supplied with the Information Disclosure Statements filed on February 28, 2002 and February 10, 2005, and for providing Applicants with an initialed copy of the PTO-1449 form filed therewith.

Applicants have also submitted the references supplied with the Information Disclosure Statement filed on September 10, 2002 for consideration by the Examiner. However, Applicants have not received an initialed copy of the PTO-1449 form indicating that the references have been considered by the Examiner. The Examiner is courteously requested to provide Applicants with an initialed copy of the PTO-1449 form filed therewith with the next official communication.

Claim Objections

Claims 1-42 have been objected to due to the presence of minor informalities. In view of the foregoing amendments, in which the Examiner's helpful suggestions have been followed, it is respectfully submitted that this objection has been addressed. Reconsideration and withdrawal of this objection are respectfully requested.

Claim Rejections Under 35 U.S.C. § 101

Claims 1-42 stand rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. This rejection is respectfully traversed.

The Examiner alleged that claims 1-42 simply manipulate abstract ideas without some claimed practical application. Applicants respectfully disagree.

Claims 1-42 are directed to a practical application within the technological arts, *i.e.*, configuring a product based on the rules in the DAG to select the components or their alternatives and to check the compatibility of the selected components/alternatives of the product.

For example, the specification on pages 18 and 39 and in FIG. 3 discloses an embodiment to configure a computer product, which includes a motherboard, a processor, two hard disks, four RAM blocks and a graphic card, etc. By using the methods of the present invention, the computer product can be configured by selecting the components (such as a motherboard, a processor, two hard disks, four RAM blocks and a graphic card) and/or their alternatives (*e.g.*, selecting a CPU from Intel Celeron processor or its alternative AMD Athlon processors) and checking the compatibility of the selected component/alternative with the other selected components based upon the rules in the DAG.

Therefore, by using the methods of claims 1-42, it can make sure that the selected components/alternatives are compatible with each other and therefore the configured computer product can work properly. Since the compatibility of the components in a computer product is always an important issue for computer manufacturing, the present invention is at least directed to, but not limited to, the application of computer manufacturing, which is a practical application

within the technological arts. In addition, the present invention is also directed to configuring other types of products based on the rules in the DAG to select the components or their alternatives and to check the compatibility of the selected components/alternatives of the product.

There are also some other discussions in the specification further illustrating that the present invention is directed to other practical applications within the technological arts. For example,

Page 17, lines 1-3: "The invention will be described in terms of a preferred implementation as applied to interactive computer assisted configuration of complex products composed of several parts, this being the origin of the problem addressed by the invention." This paragraph explicitly states that the complex product may be configured and, as will be demonstrated below, the configuration of the complex product is the first step of producing the product.

Page 17, line 23 through page 18, line 12: The product modeling is described in greater details in terms of a specific product, viz. a bike.

Page 20, lines 30-32: "The applications includes the construction of a real estate sales shop where it appears to the potential buyer of a house that he "configures" his own house". The invention as applied in accordance with this paragraph is implemented as a real estate sales shop.

Page 36, lines 28-27: "The iterative process goes on until the user decides to terminate the session. If a consistent and *complete configuration* has been found at this stage, it can be provided to an order placement system, etc." According to this paragraph, the complex configuration results in the physical ordering of the product by providing the configuration produced in accordance with the invention to an order placement system.

The texts in FIG. 3: "The system will be delivered within four working days". Accordingly, in accordance with the teachings of the present invention, a complex product may be configured, ordered and finally delivered from the manufacturer.

Since claims 1-42 are directed to at least one practical application within the technological arts, it is believed that claims 1-42 are directed to statutory subject matter and produce a useful, concrete and tangible result. Accordingly, reconsideration and withdrawal of the rejection under 35 U.S.C. § 101 are respectfully requested.

Claim Rejections Under 35 U.S.C. §§ 102 & 103

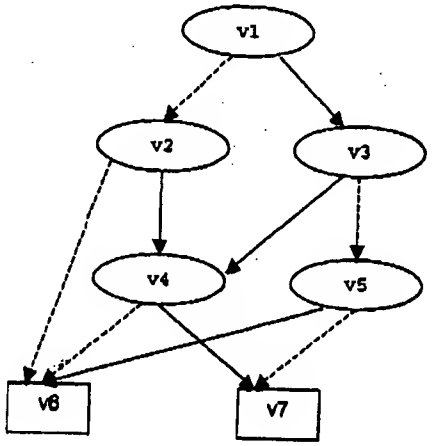
Claims 1-3, 5 and 6 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Lynch, U.S. Patent No. 5,515,524. Claims 4 and 35 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lynch in view of Polish, U.S. Patent No. 6,430,531. Claims 7-9, 24, 25, 32-34 and 36-42 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lynch in view of Henson, U.S. Patent No. 6,167,383. Claims 10-23 and 26-31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lynch in view of Henson, and further in view of Andersen, "An Introduction to Binary Decision Diagram". These rejections are respectfully traversed.

Independent claim 1 recites a combination of steps including "representing the rules in a Directed Acyclic Graph (DAG)". Applicants respectfully submit that the above combination of steps as set forth in independent claim 1 is not disclosed nor suggested by the references relied on by the Examiner.

Applicants respectfully submit that the concise definition of a directed acyclic graph (DAG) is presented in, *e.g.,* Cormen, Leiserson & Rivest "Introduction to Algorithms", MIT Press (2nd Ed., 2001) (see Attachment; hereinafter "Cormen"). Specifically, Cormen states:

> A ***directed graph*** (or ***digraph***) G is pair (V, E), where V is a finite set and E is a binary relation on V. The set V is called the ***vertex set*** of G, and its elements are called ***vertices*** (singular ***vertex***). The set E is called the ***edge set*** of G, and its elements are called ***edges***. Figure B.2(a) is a pictorial representation of a directed graph on the vertex set fl, 2, 3, 4, 5, 6). Vertices are represented by circles in the figure, and edges are represented by arrows. (Emphasis in original) (Page 1080, lines 11-16)
>
> In a directed graph, a path $(v_0, v_1, \ldots, v_k)$ formed a *cycle* if $v_0 = v_k$ and the path contains at least one edge. (Emphasis in original) (Page 1081, lines 22-23)
>
> A graph with no cycles is acyclic. (Page 1082, lines 5-6)
>
> We often take the first letters of "directed acyclic graph" and call such a graph a ***dag***. (Emphasis in original) (Page 1083, lines 15-16)

The above descriptions together define a directed, acyclic graph, often abbreviated as a DAG.



The figure above is an example of a DAG. It is *directed* (as shown by the arrowheads on the edges) and *acyclic* (no cycles, although different paths to the same *edges; e.g., the* two paths from v_1 to v_4 go through v_2 and v3 respectively).

The Examiner in the instant Office Action alleged that Lynch in FIG. 2 discloses "representing the rules in a Directed Acyclic Graph (DAG)" as recited in claim 1. Applicants respectfully disagree. In particular, Lynch in FIG. 2 merely discloses a tree structure. According to the definition stated in Cormen, FIG. 2 of Lynch is simply a connected, acyclic *undirected graph,* since there are neither directions on the edges in FIG. 2 nor any instances of two different paths to the same vertex. Therefore, the graph illustrated in FIG. 2 of Lynch is not a DAG as recited in claim 1.

In addition, the illustration of FIG. 2 of Lynch is by no means a description or a representation of a way of representing the rules. It is merely a hierarchy represented as a tree illustrating the component hierarchy (in this example the hierarchy relates to the hardware components). Therefore, Lynch fails to teach "representing the rules in a Directed Acyclic Graph (DAG)" as recited in claim 1.

With regard to the Examiner's reliance on Polish, Henson and Andersen, these references have only been relied on for their teachings related to the subject matter of dependent claims. These references also fail to disclose any Directed Acyclic Graph or the above combination of steps as set forth in independent claim 1. Accordingly, these references fail to cure the deficiencies of Lynch.

Accordingly, none of the references utilized by the Examiner individually or in combination teach or suggest the limitations of independent claim 1 or its dependent claims.

Therefore, Applicants respectfully submit that claim 1 and its dependent claims clearly define over the teachings of the references relied on by the Examiner.

Accordingly, reconsideration and withdrawal of the rejections under 35 U.S.C. §§ 102 and 103 are respectfully requested.

CONCLUSION

Since the remaining patents cited by the Examiner have not been utilized to reject the claims, but merely to show the state of the prior art, no further comments are necessary with respect thereto.

It is believed that a full and complete response has been made to the Office Action, and that as such, the Examiner is respectfully requested to send the application to Issue.

In the event there are any matters remaining in this application, the Examiner is invited to contact Joe McKinney Muncy, Registration No. 32,334 at (703) 205-8000 in the Washington, D.C. area.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37 C.F.R. §§1.16 or 1.17; particularly, extension of time fees.

Dated: November 7, 2005

Respectfully submitted,

By _____

Joe McKinney Muncy
Registration No.: 32,334
BIRCH, STEWART, KOLASCH & BIRCH, LLP
8110 Gatehouse Road
Suite 100 East
P.O. Box 747
Falls Church, Virginia 22040-0747
(703) 205-8000
Attorney for Applicant

Attachment: Cormen, Leiserson & Rivest "Introduction to Algorithms", p1080-83 (2nd Ed., 2001)

B.3-3
Give a natural definition for the inverse of a binary relation such that if a relation is in fact a bijective function, its relational inverse is its functional inverse.

B.3-4 ⋆
Give a bijection from $\mathbf{Z}$ to $\mathbf{Z} \times \mathbf{Z}$.

B.4 Graphs

This section presents two kinds of graphs: directed and undirected. Certain definitions in the literature differ from those given here, but for the most part, the differences are slight. Section 22.1 shows how graphs can be represented in computer memory.

A ***directed graph*** (or ***digraph***) G is a pair (V, E), where V is a finite set and E is a binary relation on V. The set V is called the ***vertex set*** of G, and its elements are called ***vertices*** (singular: ***vertex***). The set E is called the ***edge set*** of G, and its elements are called ***edges***. Figure B.2(a) is a pictorial representation of a directed graph on the vertex set $\{1, 2, 3, 4, 5, 6\}$. Vertices are represented by circles in the figure, and edges are represented by arrows. Note that ***self-loops***—edges from a vertex to itself—are possible.

In an ***undirected graph*** $G = (V, E)$, the edge set E consists of *unordered* pairs of vertices, rather than ordered pairs. That is, an edge is a set $\{u, v\}$, where $u, v \in V$ and $u \neq v$. By convention, we use the notation (u, v) for an edge, rather than the set notation $\{u, v\}$, and (u, v) and (v, u) are considered to be the same edge. In an undirected graph, self-loops are forbidden, and so every edge consists of exactly two distinct vertices. Figure B.2(b) is a pictorial representation of an undirected graph on the vertex set $\{1, 2, 3, 4, 5, 6\}$.

Many definitions for directed and undirected graphs are the same, although certain terms have slightly different meanings in the two contexts. If (u, v) is an edge in a directed graph $G = (V, E)$, we say that (u, v) is ***incident from*** or ***leaves*** vertex u and is ***incident to*** or ***enters*** vertex v. For example, the edges leaving vertex 2 in Figure B.2(a) are $(2, 2)$, $(2, 4)$, and $(2, 5)$. The edges entering vertex 2 are $(1, 2)$ and $(2, 2)$. If (u, v) is an edge in an undirected graph $G = (V, E)$, we say that (u, v) is ***incident on*** vertices u and v. In Figure B.2(b), the edges incident on vertex 2 are $(1, 2)$ and $(2, 5)$.

If (u, v) is an edge in a graph $G = (V, E)$, we say that vertex v is ***adjacent*** to vertex u. When the graph is undirected, the adjacency relation is symmetric. When the graph is directed, the adjacency relation is not necessarily symmetric. If v is adjacent to u in a directed graph, we sometimes write $u \rightarrow v$. In parts (a) and (b) of Figure B.2, vertex 2 is adjacent to vertex 1, since the edge $(1, 2)$ belongs to both

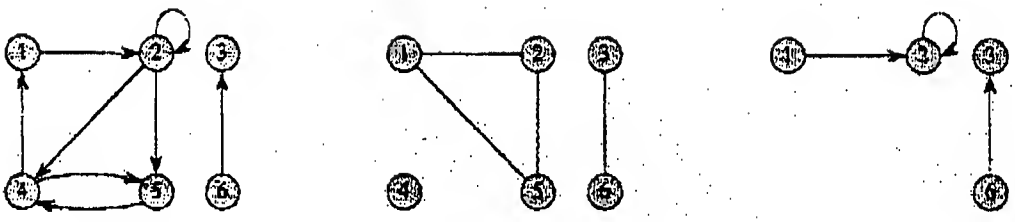


Figure B.2 Directed and undirected graphs. (a) A directed graph $G = (V, E)$, where $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (2, 2), (2, 4), (2, 5), (4, 1), (4, 5), (5, 4), (6, 3)\}$. The edge $(2, 2)$ is a self-loop. (b) An undirected graph $G = (V, E)$, where $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 5), (2, 5), (3, 6)\}$. The vertex 4 is isolated. (c) The subgraph of the graph in part (a) induced by the vertex set $\{1, 2, 3, 6\}$.

graphs. Vertex 1 is *not* adjacent to vertex 2 in Figure B.2(a), since the edge $(2, 1)$ does not belong to the graph.

The *degree* of a vertex in an undirected graph is the number of edges incident on it. For example, vertex 2 in Figure B.2(b) has degree 2. A vertex whose degree is 0, such as vertex 4 in Figure B.2(b), is *isolated*. In a directed graph, the *out-degree* of a vertex is the number of edges leaving it, and the *in-degree* of a vertex is the number of edges entering it. The *degree* of a vertex in a directed graph is its in-degree plus its out-degree. Vertex 2 in Figure B.2(a) has in-degree 2, out-degree 3, and degree 5.

A *path* of *length* k from a vertex u to a vertex u' in a graph $G = (V, E)$ is a sequence $\langle v_0, v_1, v_2, \ldots, v_k \rangle$ of vertices such that $u = v_0$, $u' = v_k$, and $(v_{i-1}, v_i) \in E$ for $i = 1, 2, \ldots, k$. The length of the path is the number of edges in the path. The path *contains* the vertices $v_0, v_1, \ldots, v_k$ and the edges $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$. (There is always a 0-length path from u to u.) If there is a path p from u to u', we say that u' is *reachable* from u via p, which we sometimes write as $u \overset{p}{\rightsquigarrow} u'$ if G is directed. A path is *simple* if all vertices in the path are distinct. In Figure B.2(a), the path $\langle 1, 2, 5, 4 \rangle$ is a simple path of length 3. The path $\langle 2, 5, 4, 5 \rangle$ is not simple.

A *subpath* of path $p = \langle v_0, v_1, \ldots, v_k \rangle$ is a contiguous subsequence of its vertices. That is, for any $0 \le i \le j \le k$, the subsequence of vertices $\langle v_i, v_{i+1}, \ldots, v_j \rangle$ is a subpath of p.

In a directed graph, a path $\langle v_0, v_1, \ldots, v_k \rangle$ forms a *cycle* if $v_0 = v_k$ and the path contains at least one edge. The cycle is *simple* if, in addition, $v_1, v_2, \ldots, v_k$ are distinct. A self-loop is a cycle of length 1. Two paths $\langle v_0, v_1, v_2, \ldots, v_{k-1}, v_0 \rangle$ and $\langle v'_0, v'_1, v'_2, \ldots, v'_{k-1}, v'_0 \rangle$ form the same cycle if there exists an integer j such that $v'_i = v_{(i+j) \bmod k}$ for $i = 0, 1, \ldots, k-1$. In Figure B.2(a), the path $\langle 1, 2, 4, 1 \rangle$

forms the same cycle as the paths $(2, 4, 1, 2)$ and $(4, 1, 2, 4)$. This cycle is simple, but the cycle $(1, 2, 4, 5, 4, 1)$ is not. The cycle $(2, 2)$ formed by the edge $(2, 2)$ is a self-loop. A directed graph with no self-loops is ***simple***. In an undirected graph, a path $\langle v_0, v_1, \ldots, v_k \rangle$ forms a ***(simple) cycle*** if $k \geq 3$, $v_0 = v_k$, and $v_1, v_2, \ldots, v_k$ are distinct. For example, in Figure B.2(b), the path $\langle 1, 2, 5, 1 \rangle$ is a cycle. A graph with no cycles is ***acyclic***.

An undirected graph is ***connected*** if every pair of vertices is connected by a path. The ***connected components*** of a graph are the equivalence classes of vertices under the "is reachable from" relation. The graph in Figure B.2(b) has three connected components: $\{1, 2, 5\}$, $\{3, 6\}$, and $\{4\}$. Every vertex in $\{1, 2, 5\}$ is reachable from every other vertex in $\{1, 2, 5\}$. An undirected graph is connected if it has exactly one connected component, that is, if every vertex is reachable from every other vertex.

A directed graph is ***strongly connected*** if every two vertices are reachable from each other. The ***strongly connected components*** of a directed graph are the equivalence classes of vertices under the "are mutually reachable" relation. A directed graph is strongly connected if it has only one strongly connected component. The graph in Figure B.2(a) has three strongly connected components: $\{1, 2, 4, 5\}$, $\{3\}$, and $\{6\}$. All pairs of vertices in $\{1, 2, 4, 5\}$ are mutually reachable. The vertices $\{3, 6\}$ do not form a strongly connected component, since vertex 6 cannot be reached from vertex 3.

Two graphs $G = (V, E)$ and $G' = (V', E')$ are ***isomorphic*** if there exists a bijection $f : V \to V'$ such that $(u, v) \in E$ if and only if $(f(u), f(v)) \in E'$. In other words, we can relabel the vertices of G to be vertices of G', maintaining the corresponding edges in G and G'. Figure B.3(a) shows a pair of isomorphic graphs G and G' with respective vertex sets $V = \{1, 2, 3, 4, 5, 6\}$ and $V' = \{u, v, w, x, y, z\}$. The mapping from V to V' given by $f(1) = u$, $f(2) = v$, $f(3) = w$, $f(4) = x$, $f(5) = y$, $f(6) = z$ is the required bijective function. The graphs in Figure B.3(b) are not isomorphic. Although both graphs have 5 vertices and 7 edges, the top graph has a vertex of degree 4 and the bottom graph does not.

We say that a graph $G' = (V', E')$ is a ***subgraph*** of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. Given a set $V' \subseteq V$, the subgraph of G ***induced*** by V' is the graph $G' = (V', E')$, where

$$E' = \{(u, v) \in E : u, v \in V'\} .$$

The subgraph induced by the vertex set $\{1, 2, 3, 6\}$ in Figure B.2(a) appears in Figure B.2(c) and has the edge set $\{(1, 2), (2, 2), (6, 3)\}$.

Given an undirected graph $G = (V, E)$, the ***directed version*** of G is the directed graph $G' = (V, E')$, where $(u, v) \in E'$ if and only if $(u, v) \in E$. That is, each undirected edge (u, v) in G is replaced in the directed version by the two directed edges (u, v) and (v, u). Given a directed graph $G = (V, E)$, the ***undirected version*** of G is the undirected graph $G' = (V, E')$, where $(u, v) \in E'$ if

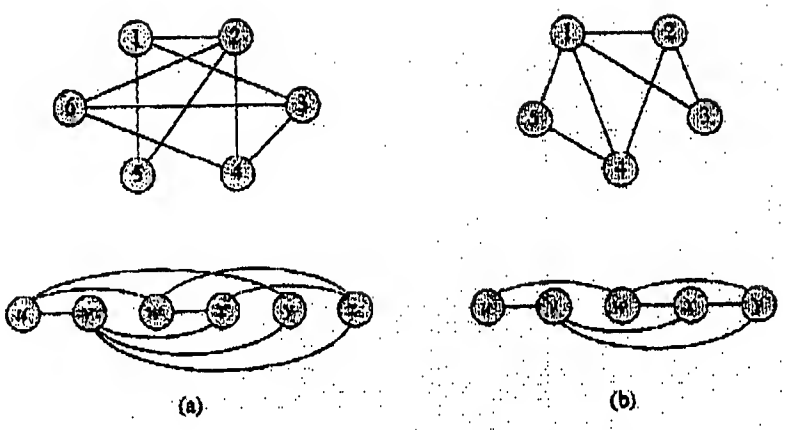


Figure B.3 (a) A pair of isomorphic graphs. The vertices of the top graph are mapped to the vertices of the bottom graph by $f(1) = u$, $f(2) = v$, $f(3) = w$, $f(4) = x$, $f(5) = y$, $f(6) = z$. (b) Two graphs that are not isomorphic, since the top graph has a vertex of degree 4 and the bottom graph does not.

and only if $u \neq v$ and $(u, v) \in E$. That is, the undirected version contains the edges of G "with their directions removed" and with self-loops eliminated. (Since (u, v) and (v, u) are the same edge in an undirected graph, the undirected version of a directed graph contains it only once, even if the directed graph contains both edges (u, v) and (v, u).) In a directed graph $G = (V, E)$, a ***neighbor*** of a vertex u is any vertex that is adjacent to u in the undirected version of G. That is, v is a neighbor of u if either $(u, v) \in E$ or $(v, u) \in E$. In an undirected graph, u and v are neighbors if they are adjacent.

Several kinds of graphs are given special names. A ***complete graph*** is an undirected graph in which every pair of vertices is adjacent. A ***bipartite graph*** is an undirected graph $G = (V, E)$ in which V can be partitioned into two sets V_1 and V_2 such that $(u, v) \in E$ implies either $u \in V_1$ and $v \in V_2$ or $u \in V_2$ and $v \in V_1$. That is, all edges go between the two sets V_1 and V_2. An acyclic, undirected graph is a ***forest***, and a connected, acyclic, undirected graph is a ***(free) tree*** (see Section B.5). We often take the first letters of "directed acyclic graph" and call such a graph a ***dag***.

There are two variants of graphs that you may occasionally encounter. A ***multigraph*** is like an undirected graph, but it can have both multiple edges between vertices and self-loops. A ***hypergraph*** is like an undirected graph, but each ***hyperedge***, rather than connecting two vertices, connects an arbitrary subset of vertices. Many algorithms written for ordinary directed and undirected graphs can be adapted to run on these graphlike structures.